AL, 57" has a machine code of B057, where BO is the opcode and 57 is the operand. Similarly, the machine code B686 is located in memory locations 1132:0102 and 1132:0103 and represents the opcode and the operand for the instruction "MOV DH,86". The physical address is an actual location within RAM (or even ROM). The following are the physical addresses and the contents of each location for the program above. Remember that it is the physical address that is put on the address bus by the 8086 CPU to be decoded by the memory circuitry.

Data segment

Assume that a program is being written to add 5 bytes of data, such as 25H, 12H, 15H, IFH, and2BH, where each byte represents a person's daily overtime pay.
One way to add them is as follows:

```
MOV   AL,00H      ;initialize AL
ADD   AL,25H      ;add 25H to AL
ADD   AL,12H      ;add 12H to AL
ADD   AL,15H      ;add 15H to AL
ADD   AL,1FH      ;add 1FH to AL
ADD   AL,2BH      ;add 2BH to AL
```

In the program above, the data and code are mixed together in the instructions.
The problem with writing the program this way is that if the data changes, the code must be searched for every place the data is included, and the data retyped. For this reason, the idea arose to set aside an area of memory strictly for data. In 80x86 microprocessors, the area of memory set aside for data is called the data segment. Just as the code segment is associated with CS and IP as its segment register and offset, the data segment uses register DS and an offset value.
The following demonstrates how data can be stored in the data segment and the program rewritten so that it can be used for any set of data. Assume that the offset for the data segment begins at 200H. The data is placed in memory locations:

```
DS:0200 = 25
DS:0201 = 12
DS:0202 = 15
DS:0203 = 1F
DS:0204 = 2B
```

and the program can be rewritten as follows:

```
MOV   AL,0          ;clear AL
ADD   AL,[0200]     ;add the contents of DS:200 to AL
ADD   AL,[0201]     ;add the contents of DS:201 to AL
ADD   AL,[0202]     ;add the contents of DS:202 to AL
ADD   AL,[0203]     ;add the contents of DS:203 to AL
ADD   AL,[0204]     ;add the contents of DS:204 to AL
```

Notice: The 8086/88 allows only the use of registers BX, SI, and DI as offset registers for the data segment. In other words, while CS uses only the IP register as an offset, OS uses only BX, DI, and SI to hold the offset address of the data. The term pointer is often used for a register holding an offset address. In the following example, BX is used as a pointer:

```
MOV   AL,0          ;initialize AL
MOV   BX,0200H      ;BX points to the offset addr of first byte
ADD   AL,[BX]       ;add the first byte to AL
INC   BX            ;increment BX to point to the next byte
ADD   AL,[BX]       ;add the next byte to AL
INC   BX            ;increment the pointer
ADD   AL,[BX]       ;add the next byte to AL
INC   BX            ;increment the pointer
ADD   AL,[BX]       ;add the last byte to AL
```

The "INC" instruction adds 1 to (increments) it~ operand. "INC BX" achieves the same result as "ADD BX,I". For the program above, if the offset address where data is located is changed, only one instruction will need to be modified and the rest of the program will be unaffected. Examining the program above shows that there is a pattern of two instructions being repeated. This leads to the idea of using a loop to repeat certain instructions. Implementing a loop requires familiarity with the flag register, discussed later in this chapter.

Logical address and physical address in the data segment

The physical address for data is calculated using the same rules as for the code segment. That is, the physical address of data is calculated by shifting OS left one hex digit and adding the offset value, as shown in Examples 1-2, 1-3, and 1-4.

Example 1-2

Assume that DS is 5000 and the offset is 1950. Calculate the physical address of the byte.

Solution:

DS				:	offset			
5	0	0	0	:	1	9	5	0

The physical address will be 50000 + 1950 = 51950.

1. Start with DS.

5	0	0	0	

2. Shift DS left.

5	0	0	0	0

3. Add the offset.

5	1	9	5	0

Example 1-3

If DS = 7FA2H and the offset is 438EH,
(a) Calculate the physical address.
(b) Calculate the lower range.
(c) Calculate the upper range of the data segment.
(d) Show the logical address.

Solution:
(a) 83DAE (7FA20 + 438E)
(b) 7FA20 (7FA20 + 0000)
(c) 8FA1F (7FA20 + FFFF)
(d) 7FA2:438E

Example 1-4

Assume that the DS register is 578C. To access a given byte of data at physical memory location 67F66, does the data segment cover the range where the data is located? If not, what changes need to be made?

Solution:
No, since the range is 578C0 to 678BF, location 67F66 is not included in this range. To access that byte, DS must be changed so that its range will include that byte.

Little endian convention:

Previous examples used 8-bit or 1-byte data. In this case the bytes are stored one after another in memory. What happens when 16-bit data is used?
For example:

```
MOV   AX,35F3H     ;load 35F3H into AX
MOV   [1500],AX    ;copy the contents of AX to offset 1500H
```

In cases like this, the low byte goes to the low memory location and the high byte goes to the high memory address. In the example above, memory location OS: 1500 contains F3H and memory location OS: 1501 contains 35H.

DS:1500 = F3 DS:1501 = 35

This convention is called little endian versus big endian. See Example 1-5.

Example 1-5

Assume memory locations with the following contents: DS:6826 = 48 and DS:6827 = 22. Show the contents of register BX in the instruction "MOV BX,[6826]".

Solution:
According to the little endian convention used in all 80x86 microprocessors, register BL should contain the value from the low offset address 6826 and register BH the value from offset address 6827, giving BL = 48H and BH = 22H.

DS:6826 = 48
DS:6827 = 22

BH	BL
22	48